# Bootstrap简介

## 概述

Bootstrap，来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。

框架:**具有一定结构和功能的代码。**

Bootstrap的学习需要先初步了解Html、Css、Javascript基础知识。

## 优缺点

优势：同样或者相似的功能，避免重复编写代码，节省代码量和开发时间。

缺点：必须按照指定的规则调用，熟悉规则需要投入时间。

## 使用场景

适合个性化定制较少的项目，例如业务系统等。

## 学习内容

1. 如何搭建环境
2. 媒体查询/响应式布局/栅格系统
3. 内置的基础样式学习
4. 常用组件学习
5. 常用插件的学习
6. 实例学习，举一反三

组件库 和 JavaScript插件集 一同提供了以下组件元素。

* 按钮组
* 按钮下拉菜单
* 用于导航的标签、pill、列表
* 导航条
* Labels
* Badges
* Page headers and hero unit
* 缩略图
* 警告对话框
* 进度条
* 模态对话框(Modals)
* 下拉菜单(Dropdowns)
* Tooltips
* Popovers
* Accordion
* Carousel
* Typeahead

## 官网地址

<http://www.bootcss.com/>

bootstrap下载

## 文件结构



初体验 感受bootstrap的强大。

开发工具下载地址：

http://www.dcloud.io/

# 全局设置

## 准备工作

### 必须使用HTML5文档类型

Bootstrap使用的某些HTML元素和CSS属性需要文档类型为HTML5 doctype。因此这一文档类型必须出现在项目的每个页面的开始部分。

1. <!DOCTYPE html>
2. <html lang="en">
3. ...
4. </html>

### 设置编码

<meta charset="utf-8">

### 引入样式表文件

<link rel="stylesheet" href="css/bootstrap.min.css">

### 引入jquery和bootstrap.min.js

<script src="js/jquery-1.11.0.js"></script>

<script src="js/bootstrap.min.js"></script>

## 样式改变

Bootstrap 为屏幕、排版和链接设置了基本的全局样式。比如：

* 移除了body的 margin
* 设置了 body 的背景颜色 background-color: white;
* 使用 @baseFontFamily、@baseFontSize 和@baseLineHeight 属性作为我们排版的基础
* 通过 @linkColor 设置全局链接颜色，并且，当链接处于 :hover 状态时才会带有下划线

### 初体验

# Emmet提高大家写代码的速度

Emmet (前身为 Zen Coding) 是一个能大幅度提高前端开发效率的一个工具。

在编辑器中输入缩写代码：ul>li\*5 ，然后按下拓展键（默认为tab），即可得到代码片段。

1 乘法 \* ul>li\*5 ，然后按下拓展键（默认为tab）

2 后代 > nav>ul>li

3 兄弟 + div+p+bq

4 上级 ^ ^div+div>p>span+em^bq

5 分组 () (div>dl>(dt+dd)\*3)+footer>p

6 自增 $ ul>li.item$\*5 倒序 ul>li.item$@-\*5 编号是3位数字 ul>li.item$$$\*5 自增从3开始 ul>li.item$@3\*5

7 ID和类 #. form#search.wide

8 自定义属性[] p[title="Hello world"] 多个属性 td[rowspan=2 colspan=3 title]

9 文本 {} a{Click me}

10隐式标签 em>.class ul>.class table>.row>.col 所有未知的缩写都会转换成标签

11 脚手架 ！

# 媒体查询技术

## 常用的媒体

all（所有）

screen（电脑 智能手机 平板）

print(打印机)

speech（应用于屏幕阅读器等发声设备）

媒体查询技术能做的事情：

同一个网页，在不同的媒体下面显示不同的样式。

## 常见的屏幕大小

* col-xs- 超小屏幕 手机(<768px)
* col-sm- 小屏幕 平板(>=768px)
* col-md- 中等屏幕 桌面显示器(>=992px)
* col-lg- 大屏幕 大桌面显示器(>=1200px)

## 根据不同的媒体定义不同的样式。

**@media 媒体类型and （媒体特性）{你的样式}**

在Media Query中如果没有明确指定Media Type，那么其默认为all

/\* 大屏幕 \*/

@media (min-width: 1200px) { ... }

/\* 平板电脑和小屏电脑之间的分辨率 \*/

@media (min-width: 768px) and (max-width: 979px) { ... }

/\* 横向放置的手机和竖向放置的平板之间的分辨率 \*/

@media (max-width: 767px) { ... }

/\* 横向放置的手机及分辨率更小的设备 \*/

@media (max-width: 480px) { ... }

## 根据不同的媒体加载不同的样式表

<link rel="stylesheet" media="screen and (max-width:480px)" href="iphone.css" />

## not和only的使用

使用关键词“not”是用来排除某种制定的媒体类型

not操作符将媒体查询的结果取反，使用not操作符必须指定设备类型，并且not操作符仅能用于一个完整的媒体查询。

only用来指定某种特定的媒体类型。

## 横向和竖向显示

orientation: landscape

orientation: portrait

/\* orientation: portrait | landscape \*/

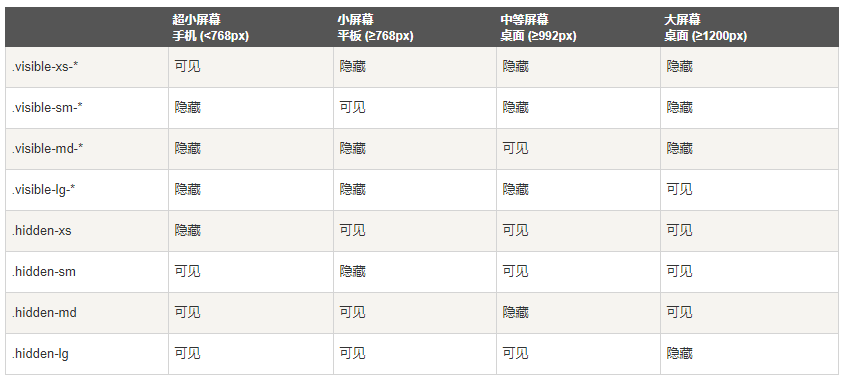
## 设备屏幕的输出宽度Device Width

在设置时，需要注意先后顺序，不然后面的会覆盖前面的样式。

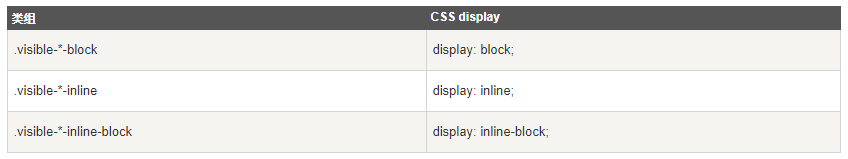
# 响应式开发及辅助类

## 可见性（显示与隐藏）

Bootstrap 提供了一些辅助类，以便更快地实现对移动设备友好的开发。这些可以通过媒体查询结合大型、小型和中型设备，实现内容对设备的显示和隐藏。



## display 属性



## 打印类

下表列出了打印类。使用这些切换打印内容。



# 栅格系统Grid System

## 使用栅格系统准备工作

Bootstrap 提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。

移动设备优先需要设置：

<meta name="viewport" content="width=device-width, initial-scale=1,maximum-scale=1, user-scalable=no">

此meta非常重要，用于设置屏幕和设备等宽以及是否运行用户缩放，及缩放比例的问题。

分别为：屏幕宽度和设备一致、初始缩放比例、最大缩放比例和禁止用户缩放。

width=device-width属性控制设备宽度，即使不同的设备带有不同的屏幕分辨率，也能保证完美的呈现。

initial-scale=1.0确保页面加载时，以1:1的比例呈现，不会有任何缩放。

user-scalable=no禁用其缩放功能

maximum-scale=1.0与user-scalable=no同时使用时，用户可以滑动屏幕。



## 栅格系统工作原理

栅格系统中，浏览器会随着屏幕的大小的增减自动分配最多12列。通过一系列的行(row)与列(column)的组合来创建页面布局。如下：

1. 行（row）必须包含在 .container （固定宽度）或 .container-fluid （100%宽度）中.

2. 通过 行（row） 在水平方向创建一组 列（column）。

3. 内容应当放置于列（column）内，并且只有列（column）可以作为行（row）的直接子元素。

4.类似 .row 和 .col-xs-4 这种预定义的类，可以用来快速创建栅格布局。  
Bootstrap 源码中定义的 mixin 也可以用来创建语义化的布局。  
5.通过为列（column）设置 padding 属性，从而创建列与列之间的间隔（gutter）。通过为 .row 元素设置负值 margin 从而抵消掉为 .container 元素设置的 padding，也就间接为“行（row）”所包含的“列（column）”抵消掉了 padding。

6.负值的 margin 就是下面的示例为什么是向外突出的原因。在栅格列中的内容排成  
一行。  
7.栅格系统中的列是通过指定 1 到 12 的值来表示其跨越的范围。例如，三个等宽的列  
可以使用三个 .col-xs-4 来创建。

8.如果一“行（row）”中包含了的“列（column）”大于 12，多余的“列（column）”  
所在的元素将被作为一个整体另起一行排列。  
9. 栅格类

练习：创建一个响应式行

|  |
| --- |
| <div class="container">  <div class="row">  ...  </div>  </div> |

//创建最多 12 列的响应式行

|  |
| --- |
| <div class="container">  <div class="row">  <div class="col-md-1 a">1</div>  <div class="col-md-1 a">2</div>  <div class="col-md-1 a">3</div>  <div class="col-md-1 a">4</div>  <div class="col-md-1 a">5</div>  <div class="col-md-1 a">6</div>  <div class="col-md-1 a">7</div>  <div class="col-md-1 a">8</div>  <div class="col-md-1 a">9</div>  <div class="col-md-1 a">10</div>  <div class="col-md-1 a">11</div>  <div class="col-md-1 a">12</div>  </div>  </div> |

//四种屏幕分类全部激活

|  |
| --- |
| <div class="container">  <div class="row">  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 a">4</div>  </div>  </div> |

## 偏移列

//有时我们可以设置列偏移，让中间保持空隙.

<div class="container">

<div class="row">

<div class="col-md-8 a">8</div>

<div class="col-md-3 col-md-offset-1 a">3</div>

</div>

</div>

## 嵌套列

//也可以嵌套，嵌满也是 12 列

|  |
| --- |
| <div class="container">  <div class="row">  <div class="col-md-9 a">  <div class="col-md-8 a">1-8</div>  <div class="col-md-4 a">9-12</div>  </div>  <div class="col-md-3 a"> 11-12 </div>  </div> |

## 列排序

|  |
| --- |
| <div class="container">  <div class="row">  <div class="col-md-9 col-md-push-3 a">9</div>  <div class="col-md-3 col-md-pull-9 a">3</div>  </div> </div> |

# 布局

## 固定布局

提供了一个通用的固定宽度(也可以变为响应式)的布局方式，仅仅用 <div class="container"> 即可。

1. <body>
2. <div class="container">
3. ...
4. </div>
5. </body>

## 流式布局

利用 <div class="container-fluid"> 代码可以创建一个流式、两列的页面 — 非常适合于应用和文档类页面。

|  |
| --- |
| <div class="container-fluid">  <div class="row-fluid">    </div>  </div> |

# 排版

Bootstrap 提供了一些常规设计好的页面排版的样式供开发者使用。

## 标题

1、HTML中定义的所有标题标签, 从<h1> 到 <h6> 都是可用的。

2、Bootstrap定义的全局 font-size 是 **14px**，line-height 是 **20px**。这些样式应用到了 <body> 和所有的段落上。另外，对 <p> (段落)还定义了1/2行高(默认为10px)的底部外边距(margin)属性。

3、通过添加 .lead 让段落突出显示。

4、对于不需要强调的inline或block类型的文本，使用small标签。

5、<strong>rendered as bold text</strong>

//从h1到h6

|  |
| --- |
| <h1>Bootstrap</h1> //36px  <h2>Bootstrap</h2> //30px  <h3>Bootstrap</h3> //24px  <h4>Bootstrap</h4> //18px  <h5>Bootstrap</h5> //14px  <h6>Bootstrap</h6> //12px |

//内联元素使用标题字体

<span class="h1">Bootstrap</span>

//在标题元素内插入 small 元素

<h1>Bootstrap 框架 <small>Bootstrap 小标题</small></h1>

<h2>Bootstrap 框架 <small>Bootstrap 小标题</small></h2>

<h3>Bootstrap 框架 <small>Bootstrap 小标题</small></h3>

<h4>Bootstrap 框架 <small>Bootstrap 小标题</small></h4>

<h5>Bootstrap 框架 <small>Bootstrap 小标题</small></h5>

<h6>Bootstrap 框架 <small>Bootstrap 小标题</small></h6>

## 内联文本元素

### 添加标记

<mark>元素或.mark 类

<p>Bootstrap<mark>框架</mark></p>

### 各种加线条的文本

<del>Bootstrap 框架</del> //删除的文本

<s>Bootstrap 框架</s> //无用的文本

<ins>Bootstrap 框架</ins> //插入的文本

<u>Bootstrap 框架</u> //效果同上，下划线文本

### 各种强调的文本

<small>Bootstrap 框架</small> //标准字号的 85%

<strong>Bootstrap 框架</strong> //加粗 700

<em>Bootstrap 框架</em> //倾斜

## 对齐类

|  |
| --- |
| //设置文本对齐  <p class="text-center">Bootstrap 框架</p> <!--居中-->  <p class="text-right">Bootstrap 框架</p> <!--居右-->  <p class="text-justify">Bootstrap 框架</p> <!--两端对齐，支持度不佳-->  <p class="text-nowrap">Bootstrap 框架</p> <!--不换行--> |

## 大小写

<p class="text-lowercase">Bootstrap 框架</p> <!--小写-->

<p class="text-uppercase">Bootstrap 框架</p> <!--大写-->

<p class="text-capitalize">bootstrap 框架</p><!--首字母大写-->

## 缩略语

当鼠标悬停在缩写和缩写词上时就会显示完整内容，Bootstrap实现了对HTML <abbr>元素的增强样式。缩略语元素带有title属性，外观表现为带有较浅的虚线框，鼠标移至上面时会变成带有“问号”的指针。

<abbr title="HyperText Markup Language" class="initialism">HTML</abbr>

## 地址

//设置地址，去掉了倾斜，设置了行高，底部 20px

|  |
| --- |
| <address>  <strong>Twitter, Inc.</strong><br>  795 Folsom Ave, Suite 600<br>  San Francisco, CA 94107<br>  <abbr title="Phone">P:</abbr> (123) 456-7890  </address> |

## 引用选项

<blockquote>

一个带有源标题的引用

<small>Someone famous in <cite title="Source Title">Source Title</cite>

</small>

</blockquote>

<blockquote class="pull-right">

一个向右对齐的标题引用

<small>

Someone famous in

<cite title="Source Title">Source Title</cite>

</small>

</blockquote>

## 列表

### 移除默认样式

|  |
| --- |
| <ul class="list-unstyled">  <li>Bootstrap 框架</li>  <li>Bootstrap 框架</li>  </ul> |

### 设置为内联

|  |
| --- |
| <ul class="list-inline">  <li>Bootstrap 框架</li>  <li>Bootstrap 框架</li>  <li>Bootstrap 框架</li>  <li>Bootstrap 框架</li>  <li>Bootstrap 框架</li>  </ul> |

### 水平描述

使<dl>中的每个条目和其描述一对一水平显示。

|  |
| --- |
| <dl class="dl-horizontal">  <dt>Bootstrap</dt>  <dd>Bootstrap 提供了一些常规设计好的页面排版的样式供开发者使用。</dd>  </dl> |

## 代码

Bootstrap允许使用两种方式显示代码:

1.第一种是<code>标签。如果想要内联显示代码，就应该使用<code>标签

2.第二种是<pre>标签。如果想要用独立的块元素显示代码，则使用<pre>标签

3.开始和结束标签，请使用unicode变体: &lt(<);和&gt(>)

风格1：通过<code>标签内嵌一小段代码。

|  |
| --- |
| <code>&lt;section&gt;</code> should be wrapped as inline. |

风格2：使用<pre>展示多行代码。为了能够正确展示，务必将代码中的任何尖括号做转义。

|  |
| --- |
| <pre>  &lt;p&gt;Sample text here...&lt;/p&gt;  </pre> |

扩展： kbd显示代码

<kbd>ctrl + ,</kbd>

# 表格

## 受支持的表格标签

受支持的表格标签列表以及使用方法。

| **标签** | **描述** |
| --- | --- |
| <table> | 包裹以表格形式展示数据的元素 |
| <thead> | 包含表头(<tr>) 的容器 |
| <tbody> | 包含表格内容(<tr>)的容器 |
| <tr> | 单元格 (<td> 或 <th>) 容器 |
| <td> | 默认的单元格 |
| <th> | 每列(或行，依赖于放置的位置)所对应的的label |
| <caption> | 用于对表的描述或总结，对屏幕阅读器特别有用 |

1.(.table): 为任意<table>添加基本样式(只有横向分割线)

2.(.table-striped): 为<tbody>内添加斑马线形式条纹

3.(.table-bordered): 为所有表格的单元格添加边框

4.(.table-hover): 在<tbody>内的任一行启用鼠标悬停状态

5.(.table-condensed): 让表格更加紧凑

<tr>、<th>、<td>类

下列类可用于表格的行或单元格

1.(.active): 将悬停的颜色应用于行或单元格

2.(.success): 表示成功的操作

3.(.info): 表示信息变化的操作

4.(.warning): 表示一个警告的操作

5.(.danger): 表示一个危险的操作

通过将任意的.table包在.table-responsive内，可以让表格滚动，以适应小型设备和大型设备

## 代码素材

|  |
| --- |
| <table>  <tr><th>姓名</th><th>性别</th><th>年龄</th><th>学号</th><th>专业</th></tr>  <tr><td>张三</td><td>男</td><td>18</td><td>20180101</td><td>数学与应用数学</td></tr>  <tr><td>李四</td><td>女</td><td>19</td><td>20180102</td><td>计算机科学与技术</td></tr>  <tr><td>王五</td><td>男</td><td>18</td><td>20180103</td><td>大数据</td></tr>  <tr><td>赵六</td><td>男</td><td>18</td><td>20180104</td><td>物联网</td></tr>  <tr><td>李冰</td><td>男</td><td>17</td><td>20180105</td><td>物联网</td></tr>  </table> |

## 默认样式

为 <table> 标签增加基本样式--很少的内补(padding)并只增加水平分隔线--只要为其增 .table 类即可。

## 隔行换色 .table-striped

## 增加边框和圆角 .table-bordered

为表格增加边框(border)和圆角(rounded corner)。

## .table-hover

为 <tbody> 中的每一行赋予鼠标悬停样式。

## .table-condensed

每个单元格的内补(padding)减半可使表格更紧凑。

## 可选的行属性

选择情景(contextual)类为表格添加颜色。

| **Class** | **Description** |
| --- | --- |
| .success | 表示成功或积极的行为。 |
| .error | 表示一个危险或存有潜在危险的行为。 |
| .warning | 表示警告，可能需要注意。 |
| .info | 作为一个默认样式的一个替代样式。 |

# 表单

Bootstrap提供了下列类型的表单布局：

1.垂直表单(默认)

2.内联表单

3.水平表单

单独的表单控件会被自动赋予一些全局样式。所有设置了 .form-control 类的 <input>、<textarea> 和 <select> 元素都将被默认设置宽度属性为 width: 100%;。 将 label 元素和控件包裹在 .form-group 中可以获得最好的排列。

## （垂直）基本表单

基本的表单结构是Bootstrap自带的，下面列出创建一个基本表单的步骤：

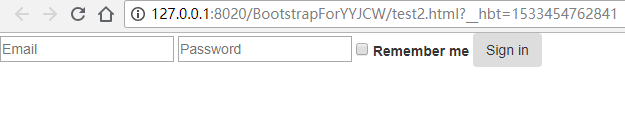
1.把标签和控件放在一个带有class .form-group的<div>中。这是获取最佳间距所必需的

2.向所有的文本元素<input>、<textarea>和<select>添加class="from-control"



|  |
| --- |
| <form>  <div class="form-group">  <label>电子邮件</label>  <input type="email" class="form-control" placeholder="请输入您的电子邮件">  </div>  <div class="form-group">  <label>密码</label>  <input type="password" class="form-control" placeholder="请输入您的密码"> </div>  </form> |

## 内联表单



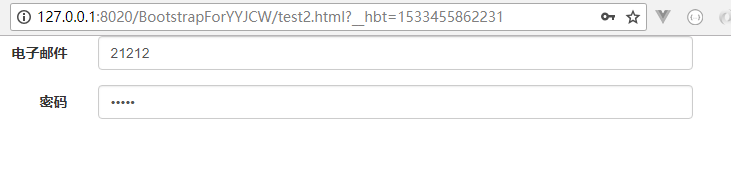
创建一个表单，希望它的所有元素都是内联的，向左对齐的，标签是并排的。请在<form>标签中添加 class .form-inline

|  |
| --- |
| <div class="container">  <form action="" method="" class="form-inline">  <div class="form-group">  <label>用户名</label><input type="text" placeholder="请输入用户名..." class="form-control" name="username" id="username" value="" />  </div>  <div class="form-group">  <label>密码</label><input type="password" placeholder="请输入密码..." class="form-control" name="password" id="password" value="" />  </div>  <button type="button" class="btn btn-default">登录</button>  </form>  </div> |

## 水平表单

水平表单与其他表单不仅标记的数量上不同，而且表单的呈现形式也不同。如需创建一个水平布局的表单，请按下面的几个步骤进行：

* 向父 <form> 元素添加 class *.form-horizontal*。
* 把标签和控件放在一个带有 class *.form-group* 的 <div> 中。
* 向标签添加 class *.control-label*。



通过为表单添加 .form-horizontal 类，并联合使用 Bootstrap 预置的栅格类，可以将 label 标签和控件组水平并排布局。这样做将改变 .form-group 的行为，使其表现为栅格系统中的行（row），因此就无需再额外添加 .row 了。

|  |
| --- |
| <form class="form-horizontal">  <div class="form-group">  <label class="col-sm-2 control-label">电子邮件</label>  <div class="col-sm-10">  <input type="email" class="form-control" placeholder="请输入您的电子邮件">  </div>  </div>  <div class="form-group">  <label class="col-sm-2 control-label">密码</label>  <div class="col-sm-10">  <input type="password" class="form-control" placeholder="请输入您的密码">  </div>  </div>  </form> |

## 表单组合

//前后增加片段

<div class="input-group">

<div class="input-group-addon">￥</div>

<input type="text" class="form-control">

<div class="input-group-addon">.00</div>

</div>



## 支持的表单控件

Bootstrap 支持最常见的表单控件，主要是 *input、textarea、checkbox、radio 和 select*。

### 输入框（Input）

最常见的表单文本字段是输入框 input。用户可以在其中输入大多数必要的表单数据。Bootstrap 提供了对所有原生的 HTML5 的 input 类型的支持，包括：*text、password、datetime、datetime-local、date、month、time、week、number、email、url、search、tel* 和 *color*。适当的 *type* 声明是必需的，这样才能让 *input* 获得完整的样式。

### 多行文本：文本域（Textarea）

此表单控件支持多行文本。可根据需要修改rows属性。

<textarea rows="3" class="form-control"></textarea>

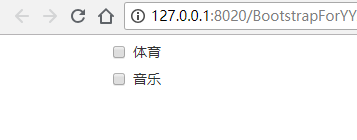
### 复选框和单选框

1.checkbox提供用于从列表中选择若干个选项，而radio限制用户只能选择一个选项

2.使用class .checkbox-inline或.radio-inline，控制它们显示在同一行.

复选框是用于在一个列表中选择一个或多个选项，而单选框是在众多选择中选择一个选项。

#### 默认（堆叠式）



|  |
| --- |
| <div class="checkbox">  <label> <input type="checkbox">体育 </label>  </div>  <div class="checkbox">  <label> <input type="checkbox">音乐 </label>  </div> |

#### 行内复选框

为复选框或单选框控件添加.inline类，使他们排列在同一行。



|  |
| --- |
| <label class="checkbox-inline">  <input type="checkbox">体育  </label>  <label class="checkbox-inline disabled">  <input type="checkbox" >音乐  </label> |

### 下拉框

选择框允许用户从多个选项中进行选择，默认情况下只能选择一个选项

1.使用<select>展示列表选项，通常是用户熟悉的选择列表

2.使用multiple="multiple"允许用户选择多个选项



可使用默认的选项或者指定multiple="multiple"属性一次显示多个选项。

<select class="form-control">

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

<option>5</option>

</select>

### 表单控件状态

1.输入框焦点：当输入框(input)接收到focus时，输入框的轮廓会被移除，同时应用box-shadow

2.禁用的输入框input：使用属性disabled禁用输入框。

3.禁用的字段集fieldset：对<fieldset>添加disabled属性来禁用<fieldset>内的所有控件

4.验证状态：对父元素简单的添加适当的class(.has-warning、has-error或.has-success)即可验证.

|  |
| --- |
| <form role="form" class="form-horizontal">  <div class="form-group">  <label class="col-sm-2 control-label">聚焦</label>  <div class="col-sm-10">  <input type="text" class="form-control" placeholder="请输入框获得焦点...">  </div>  </div>  <div class="form-group">  <label class="col-sm-2 control-label">禁用</label>  <div class="col-sm-10">  <input type="text" class="form-control" placeholder="该输入框禁止输入..." disabled>  </div>  </div>  <fieldset disabled>  <div class="form-group">  <label class="col-sm-2 control-label">禁止选择菜单</label>  <div class="col-sm-10">  <select class="form-control">  <option>禁止选择</option>  </select>  </div>  </div>  </fieldset>  <div class="form-group has-error">  <label class="col-sm-2 control-label" for="inputWarning">输入告警</label>  <div class="col-sm-10">  <input type="text" class="form-control" id="inputWarning">  </div>  </div>  </form> |

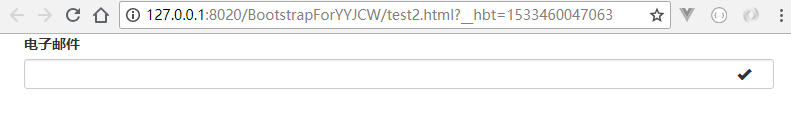
<div class="form-group has-error">

任何包含在此元素之内的 .control-label、.form-control 和 .help-block 元素都将接受这些校验状态的样式。

//label 标签同步相应状态

<label class="control-label">Input with success</label>

### 添加额外的图标



<div class="form-group has-feedback">

<label>电子邮件</label>

<input type="email" class="form-control">

<span class="glyphicon glyphicon-ok form-control-feedback"></span>

</div>

注：除了 glyphicon-ok 外，还有几个如下表：  
样式 说明  
glyphicon-ok 成功状态  
glyphicon-warning-sign 警告状态  
glyphicon-remove 错误状态

### 控件尺寸大小

使用class .input-lg和.col-lg-\* 来设置表单的高度和宽度。

1.input-lg: 应用在input上, 放大表单高度。

2.input-sm: 应用在input上, 缩小表单高度。

3.col-lg-\*: 这个样式应用在div上面，表示在大屏幕上div占用几个格子。

如: col-sm-10表示在小屏幕该div占用10个列的宽度

//从大到小

<input type="password" class="form-control input-lg">

<input type="password" class="form-control">

<input type="password" class="form-control input-sm">

注：也可以设置父元素 form-group-lg、form-group-sm，来调整。

### 表单帮助文本(help-block)

表单控件可以在输入框input上有一个块级帮助文本，使用.help-block来占用整个宽度的内容块

<form role="form">

<span>帮助文本实例</span>

<input class="form-control" type="text" placeholder="">

<span class="help-block">说明文档</span>  
</form>

# 按钮

## 按钮样式

任何带有class .btn的元素都会继承圆角灰色按钮。下列样式同样适用于<a>、<button>和<input>

1.btn：为按钮添加基本样式

2.btn-default：默认／标准按钮

3.btn-success(primary/info/warning/danger)：表示成功等的动作

4.btn-link：让按钮看起来像链接(仍然保留按钮行为)

5.btn-lg(btn-sm/xs)：制作一个大按钮等

6.btn-block：块级按钮

7.active：按钮被点击

8.disabled：禁用按钮

<button type="button" class="btn btn-info">警告按钮</button>

## 按钮状态

Bootstrap提供了激活、禁用等按钮状态的class

1.激活状态(active)：激活状态呈现被压的外观(深色的背景、深色的边框、阴影)

2.禁用状态(disabled)：禁用一个按钮时，它的颜色会变淡50%，并失去渐变。

<p>

<button type="button" class="btn btn-default">默认按钮</button>

</p>

<p>

<button type="button" class="btn btn-default active">激活按钮</button>

</p>

<p>

<button type="button" class="btn btn-default disabled">禁用按钮</button>

</p>

# 图片

为<img>元素添加相应的类就可以很容易的给图片设置样式。

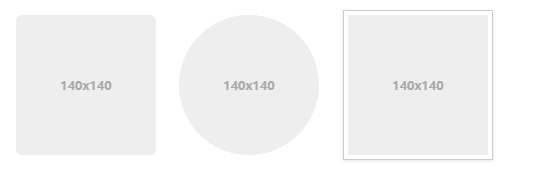
Bootstrap提供了三个对图片应用的简单样式的class：

1.img-rounded：将图片变成圆角，通过添加border-radius:6px来实现。

2.img-circle：将图片变成圆形，通过添加border-radius：50%来实现。

3.img-thumbnail：添加内边框效果，通过内边距(padding)和一个灰色的边框来实现。

4.img-responsive：让图片支持响应式设计。图片可以很好的扩展到父元素。使用max-width:100%和height:auto来实现



## 图片形状

//三种形状

<img src="img/pic.png" alt="图片" class="img-rounded">

<img src="img/pic.png" alt="图片" class="img-circle">

<img src="img/pic.png" alt="图片" class="img-thumbnail">

## 响应式图片

<img src="img/pic.png" alt="图片" class="img-responsive">

当图片大于父元素的宽度时,图片会缩小到父元素的宽度  
max-width: 100%。

# Bootstrap 辅助类

Bootstrap定义了诸多辅助类

## 文本

不同的类展示不同的文本颜色。包括链接标签<a>文本也起同样的效果

1.text-muted：文字变弱的样式

2.text-primary

3.text-success

4.text-info

5.text-warning

6.text-danger

代码：

<div class="container">

<p class="text-danger">使用text-danger样式文本</p>

<a class="text-danger" href="https://yyjcw.com ">链接文本使用text-danger</a>

</div>

## 背景

不同的类展示不同的背景颜色。如果是链接，鼠标移上去文本会变暗

1.bg-primary

2.bg-success

3.bg-info

4.bg-warning

5.bg-danger

代码:

<div class="container">

<p class="bg-danger">使用bg-danger样式文本</p>

<a class="bg-danger" href="http://write.blog.csdn.net/postlist">链接文本使用bg-danger</a>

</div>

## 其他

1.pull-left：元素浮动在左边

2.pull-right：元素浮动在右边

3.center-block：设置元素为display:block并居中显示

4.clearfix：清除浮动

5.show：强制元素显示

6.hidden：强制元素隐藏

7.sr-only：除了屏幕阅读器外，其他设备上隐藏元素

8.sr-only-focusable：与 .sr-only 类结合使用，在元素获取焦点时显示(如：键盘操作的用户)

9.text-hide：将页面元素所包含的文本内容替换为背景图

10.close：显示关闭按钮

11.caret：显示下拉式功能

## 关闭标签

使用class close得到关闭标签。

代码:

<button type="button" class="close" aria-hidden="true">&times;</button>

===========布局组件=============

# Bootstrap 字体图标(Glyphicons)

主要应用在按钮、工具条中的按钮组、导航或输入框等地方。

## 用法

如需要使用图标，只需使用下列代码即可。

代码

<span class="glyphicon glyphicon-user"></span>

## 定制字体图标和尺寸

代码:

<button type="button" class="btn btn-primary btn-lg">

<span class="glyphicon glyphicon-user"></span>User

</button>

# Bootstrap 下拉菜单(Dropdowns)

Bootstrap下拉菜单是可切换的，是以列表格式显示链接的上下文菜单

## 代码

|  |
| --- |
| <div class="dropdown">  <button type="button" class="btn btn-default" data-toggle="dropdown">专业选择 <span class="caret"></span></button>  <ul class="dropdown-menu">  <li class="dropdown-header">第一组</li>  <li><a href="###">listitem1</a></li>  <li><a href="###">listitem2</a></li>  <li class="divider"></li>  <li class="dropdown-header">第二组</li>  <li><a href="###">listitem3</a></li>  <li><a href="###">listitem4</a></li>  <li class="disabled"><a href="###">listitem5</a></li>  </ul>  </div> |

## 选项

1.dropdown-header(标题)：使用dropdown-header向下拉菜单的标签区域添加标题

2.dropup：指定向上弹出的下拉菜单

3.disabled：下拉菜单中的禁用项

4.divider：下拉菜单中的分割线

# Bootstrap 按钮组

## 按钮组介绍

通过按钮组容器把一组按钮放在同一行里。

1.btn-group：用于形成基本的按钮组，在btn-group中放置一系列带有class .btn按钮

2.btn-toolbar：用于将几组<div class="btn-group">结合在一起

3.btn-group-lg(btn-group-sm,btn-group-xs)：整个按钮组的大小控制

4.btn-group-vertical：让一组按钮垂直堆叠显示

## 基本按钮组(btn-group)

btn-group用于形成按钮组，位于同一行中。

代码：

<div class="btn-group">

<button type="button" class="btn btn-default">按钮1</button>

<button type="button" class="btn btn-danger">按钮2</button>

<button type="button" class="btn btn-info">按钮3</button>

</div>

## 按钮工具栏(btn-toolbar)

btn-toolbar用于将几个按钮组合并在一起

代码:

<div class="btn-toolbar" role="toolbar">

<div class="btn-group">

<button type="button" class="btn btn-default">按钮1</button>

<button type="button" class="btn btn-default">按钮2</button>

</div>

<div class="btn-group">

<button type="button" class="btn btn-default">按钮3</button>

<button type="button" class="btn btn-default">按钮4</button>

</div>

</div>

## 按钮的大小(btn-group-sm)

btn-group-sm用于控制按钮大小，如：<div class="btn-group btn-group-sm">

## 按钮组嵌套

可以在一个按钮组内嵌套另一个按钮组，比如：下拉菜单和一系列的按钮组一起使用时候。

代码:

<div class="btn-group">

<button type="button" class="btn btn-default">按钮1</button>

<button type="button" class="btn btn-default">按钮2</button>

<div class="btn-group">

<button type="button" class="btn btn-info dropdown-toggle" data-toggle="dropdown">

<span>下拉</span>

<span class="caret"></span>

</button>

<ul class="dropdown-menu" role="menu">

<li><a href="#">下拉链接 1</a></li>

<li><a href="#">下拉链接 2</a></li>

</ul>

</div>

</div>

## 垂直的按钮组(btn-group-vertical)

使用btn-group-vertical来让按钮垂直显示

## Bootstrap 按钮下拉菜单

## 分割的按钮下拉菜单(divider)

使用class .divider来实现分割下拉。代码: <li class="divider"></li>

## 按钮下拉菜单的大小(btn-lg)

使用class .btn-lg、btn-sm和btn-xs来指定按钮大小

代码：

<div class="btn-group">

<button type="button" class="btn btn-default btn-lg dropdown-toggle" data-toggle="dropdown">

<span>默认下拉菜单</span>

<span class="caret"></span>

</button>

<ul class="dropdown-menu" role="menu">

<li><a href="#">功能1</a></li>

<li><a href="#">功能2</a></li>

<li class="divider"></li>

<li><a href="#">分割线后的功能1</a></li>

</ul>

</div>

## 按钮上拉菜单(dropup)

菜单也可以往上拉，只需要在class .btn-group中添加class .dropup

# Bootstrap 导航元素

Bootstrap提供用于定义导航元素的一些选项。使用相同的标记和基类.nav

## 页签导航菜单(nav-tabs)

创建一个标签式的导航菜单：

1.以一个带有class .nav的无序列表开始

2.添加class .nav-tabs

代码:

<ul class="nav nav-tabs">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">iMac</a></li>

<li><a href="#">iPhone</a></li>

<li><a href="#">Mac</a></li>

</ul>

## 胶囊式的导航菜单(nav-pills)

如果需要将标签改成胶囊式，只需要将class .nav-pills替换class .nav-tabs即可

代码:

<ul class="nav nav-pills">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">iMac</a></li>

<li><a href="#">iPhone></a></li>

<li><a href="#">Mac</a></li>

</ul>

## 垂直的胶囊式导航菜单(nav-stacked)

在使用calss .nav和class .nav-pills的同时使用class .nav-stacked，让胶囊垂直堆叠

代码:

<ul class="nav nav-pills nav-stacked">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">iPad</a></li>

<li><a href="#">iMac</a></li>

<li><a href="#">iPhone</a></li>

</ul>

## 两端对齐的导航(nav-justified)

当屏幕宽度大于768px时，在使用表格式导航(nav nav-tabs)和胶囊式导航(nav nav-pills)时，请同时使用class .nav-justified。以让导航的长度和父元素等宽。

## 禁用链接(disabled)

对每个class .nav添加class .disabled，则会创建一个灰色的链接

代码:

<ul class="nav nav-tabs nav-justified">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">iPad</a></li>

<li><a href="#">iPhone</a></li>

<li class="disabled"><a href="#">iMac(禁用链接)</a></li>

<li><a href="#">iWatch</a></li>

</ul>

## 带有下拉菜单的标签

向一个标签添加下拉菜单的步骤如下:

1.以一个带有class .nav的无序列表开始

2.添加class nav-tabs

3.添加带有.dropdown-menu的无序列表

代码：

<ul class="nav nav-tabs nav-justified">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">iMac</a></li>

<li><a href="#">iPad</a></li>

<li class="dropdown">

<a class="dropdown-toggle" data-toggle="dropdown" href="#">

<span>iPhone</span>

<span class="caret"></span>

</a>

<ul class="dropdown-menu" role="menu">

<li><a href="#">iPhone-5</a></li>

<li><a href="#">iPhone-6</a></li>

<li><a href="#">iPhone-6s</a></li>

<li class="divider"></li>

<li><a href="#">iPhone-X</a></li>

</ul>

</li>

<li><a href="#">iWatch</a></li>

</ul>

## 动态标签

与tab-pane和data-toggle="tab"(或data-toggle="pill")一同使用，设置标签页对应的内容随标签切换而更改

代码：

<ul class="nav nav-tabs nav-justified">

<li class="active"><a data-toggle="tab" href="#home" >Home</a></li>

<li><a data-toggle="tab" href="#iMac">iMac</a></li>

<li><a data-toggle="tab" href="#iPhone">iPhone</a></li>

<li><a data-toggle="tab" href="#iWatch">iWatch</a></li>

</ul>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<span>首页，欢迎你</span>

</div>

<div id="iMac" class="tab-pane fade">

<span>iMac页面</span>

</div>

<div id="iPhone" class="tab-pane fade">

<span>iPhone页面</span>

</div>

<div id="iWatch" class="tab-pane fade">

<span>iWatch页面</span>

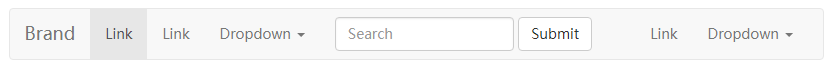
</div>

</div>

# Bootstrap 导航栏

Bootstrap导航栏包含了站点名称和基本导航定义样式

## 默认的导航栏



创建一个默认导航栏的基本步骤:

1.向<nav>标签添加 navbar和 navbar-default

2.向上面的元素添加role="navigation"，有助于增加可访问性

3.向<div>元素添加一个标题navbar-header，内部包含了带有navbar-brand的元素<a>

4.向导航栏添加链接，添加 nav和navbar-nav的无序列表

代码:

<nav class="navbar navbar-default" role="navigation">

<div class="navbar-header">

<a class="navbar-brand" href="#">Apple公司</a>

</div>

<div>

<ul class="nav navbar-nav">

<li class="active"><a href="#">iMac</a></li>

<li><a href="#">iPad</a></li>

<li class="dropdown">

<a data-toggle="dropdown" href="#" class="dropdown-toggle">

<span>iPhone</span>

<span class="caret"></span>

</a>

<ul class="dropdown-menu" role="menu">

<li><a href="#">iPhone-5s</a></li>

<li><a href="#">iPhone-6s</a></li>

<li><a href="#">iPhone-7</a></li>

<li><a href="#">iPhone-X</a></li>

</ul>

</li>

<li><a href="#">iWatch</a></li>

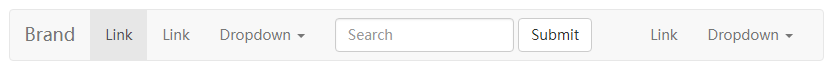
</ul>

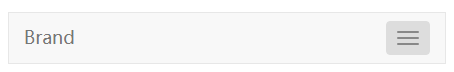
</div>

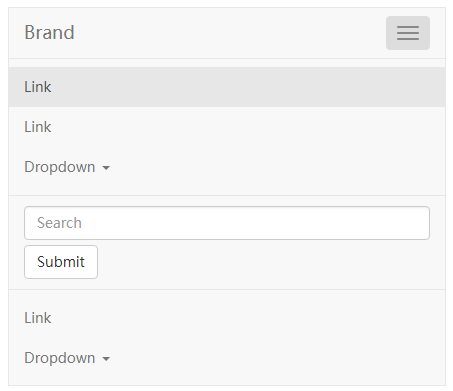
</nav>

## 响应式的导航栏

响应式的导航栏是指点击某按钮后，展开菜单。







代码:

<nav class="navbar navbar-default" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse"

data-target="#example-navbar">

<span class="sr-only">切换导航</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a href="#" class="navbar-brand">Apple公司</a>

</div>

<div class="collapse navbar-collapse" id="example-navbar">

<ul class="nav navbar-nav">

<li class="active"><a href="#">iMac</a></li>

<li><a href="#">iPad</a></li>

<li class="dropdown">

<a href="#" class="dropdown-toggle" data-toggle="dropdown">

<span>iPhone</span>

<span class="caret"></span>

</a>

<ul class="dropdown-menu" role="menu">

<li><a href="#">iPhone-5</a></li>

<li><a href="#">iPhone-6s</a></li>

<li><a href="#">iPhone-7</a></li>

<li><a href="#">iPhone-8</a></li>

<li class="divider"></li>

<li><a href="#">iPhone-X</a></li>

</ul>

</li>

</ul>

</div>

</div>

</nav>

## 导航栏中的表单(navbar-form)

导航栏中的表单使用class .navbar-form，其确保了表单适当的垂直对齐和在较窄的窗口中折叠行为。

代码:

<nav class="navbar navbar-default" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Apple公司</a>

</div>

<form class="navbar-form navbar-left" role="search">

<div class="form-group">

<input type="text" class="form-control" placeholder="输入手机型号">

</div>

<button type="submit" class="btn btn-default">提交</button>

</form>

</div>

</nav>

## 导航栏中的按钮(navbar-btn)

可以使用class .navbar-btn向不在<form>元素中添加<button>按钮

代码:

<nav class="navbar navbar-default" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Apple公司</a>

</div>

<form class="navbar-form navbar-left" role="search">

<div class="form-group">

<input type="text" class="form-control" placeholder="输入手机型号">

</div>

<button type="submit" class="btn btn-default">提交</button>

</form>

<button type="button" class="btn btn-default navbar-btn">表单外的Button</button>

</div>

</nav>

## 导航栏中的文本(navbar-text)

使用class .navbar-text向导航栏中添加文本字符串。通常与<p>标签一起使用，确保适当的前导和颜色

代码:

<nav role="navigation" class="navbar navbar-default">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Apple公司</a>

</div>

<div class="navbar-text">

<p>我是.</p>

</div>

</div>

</nav>

## 结合图标的导航链接

在常规的导航栏中使用常用的图标，更多图标请参见[图标地址](http://www.runoob.com/bootstrap/bootstrap-glyphicons.html)

代码：

<nav class="navbar navbar-default" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<a href="#" class="navbar-brand">Apple 公司</a>

</div>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-user"></span>注册</a></li>

<li><a href="#"><span class="glyphicon glyphicon-log-in"></span>登录</a></li>

</ul>

</div>

</nav>

## 组件对齐方式

1.navbar-left：向左对齐导航栏中的元素

2.navbar-right：向右对齐导航栏中的元素

## 固定到顶部(navbar-fixed-top)

Bootstrap导航栏可以动态定位。导航栏固定在顶部，让它成为随着页面一起滚动的静态导航栏。使用class .navbar-fixed-top。如：<nav class="navbar navbar-default navbar-fixed-top" role="navigation">

## 固定到底部(navbar-fixed-bottom)

使用class .navbar-fixed-bottom来让导航栏固定在页面底部

## 静态的顶部(navbar-static-top)

使用class .navbar-static-top，创建随着页面一起滚动的导航栏

## 反色的导航栏(navbar-inverse)

使用navbar-inverse创建一个带有黑色背景白色文本的反色导航栏

# Bootstrap 面包屑导航(Breadcrumbs)

Bootstrap面包屑导航是一个简单的带有.breadcrumb class无序列表

代码:

<ul class="breadcrumb">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">2018</a></li>

<li><a href="#">03</a></li>

<li><a href="#">17</a></li>

</ul>

# Bootstrap 分页

Bootstrap提供多个处理分页的class

1.pagination: 添加该class来在页面上显示分页

2.disabled: 定义当前链接不可点

3.active: 定义当前链接激活

4.pagination-lg(pagination-sm): 获取不同的大小项

## 默认的分页(pagination)

使用pagination来实现分页

代码:

<ul class="pagination">

<li><a href="#">&laquo;</a></li>

<li class="active"><a href="#">1</a></li>

<li class="disabled"><a href="#">2</a></li>

<li><a href="#">3</a></li>

<li><a href="#">&raquo;</a></li>

</ul>

## 分页的状态(disabled、active)

disabled禁用当前链接，active激活当前链接。代码如上

## 分页的大小(pagination-lg)

使用pagination-lg将会放大当前链接。使用代码:<ul class="pagination pagination-lg">

## 翻页(pager)

如果只想提供上一页／下一页，使用翻页就能满足。

1.pager: 添加该class来获得翻页链接

2.previous：把链接向左对齐

3.next：把链接向右对齐

4.disabled: 禁用当前链接

代码：

<ul class="pager">

<li class="previous disabled"><a href="#">&larr; Older</a></li>

<li class="next"><a href="#">Newer &rarr;</a></li>

</ul>

# Bootstrap 标签(label)

使用class .label来显示标签。标签可用于计数、提示和页面上其他的标记显示

1.label label-default: 默认的灰色按钮

2.label label-primary: 蓝色按钮

3.label label-success: 绿色按钮

代码:

<span class="label label-default">默认标签</span>

<span class="label label-primary">主要标签</span>

<span class="label label-success">成功标签</span>

<span class="label label-info">信息标签</span>

<span class="label label-warning">警告标签</span>

<span class="label label-danger">危险标签</span>

# Bootstrap 徽章(Badges)

徽章和标签相似，主要区别是徽章的边角更加圆滑。徽章常被用于突出显示新的或未读的项

代码：

<div class="container">

<h2>徽章</h2>

<p>

<a href="#">收件箱<span class="badge">3</span></a>

</p>

</div>

## 激活导航状态

可以将徽章放置在导航表中

代码：

<ul class="nav nav-pills">

<li class="active"><a href="#">好友数目<span class="badge">3</span></a></li>

<li><a href="#">请求加您为好友<span class="badge">10</span></a></li>

</ul>

# Bootstrap 巨幕(jumbotron)

超大屏幕可以增加标题的大小，并为登陆页面内容添加更多的外边距。使用步骤如下:

1.创建一个带有jumbotron样式的容器<div>

2.除了更大的<h1>，字体粗细font-weight被减为200

代码:

<div class="container">

<div class="jumbotron">

<h1>欢迎登陆页面</h1>

<p>这是一个超大屏幕(jumbotron)的实例</p>

<p><a class="btn btn-info btn-lg" role="button">学习更多</a></p>

</div>

</div>

## 非圆角页面

为了占用全部宽度且不带圆角的超大屏幕，请在class .container外使用class .jumbotron

代码:

<div class="jumbotron">

<div class="container">

<h1>欢迎登陆页面</h1>

<p>这是一个超大屏幕(jumbotron)的实例</p>

<p><a class="btn btn-info btn-lg" role="button">学习更多</a></p>

</div>

</div>

# Bootstrap 页面标题(page-header)

当一个网页中有多个标题且每个标题之间需要添加一定的间距时，页面标题这个功能就特别有用。

代码：

<div class="page-header">

<h1>

页面标题实例

<small>子标题</small>

</h1>

</div>

<p>内容部分</p>

# Bootstrap 缩略图(thumbnail)

Bootstrap使用缩略图可为大多数站点中布局图像、视频、文本等。具体步骤如下:

1.在图像周围添加带有class .thumbnail的<a>标签

2.这会添加四个像素的内边距(padding)和一个灰色按钮

3.当鼠标悬停在图像上，会动画的显示出图像的轮廓

代码:

<div class="row">

<div class="col-sm-6 col-md-3">

<a href="#" class="thumbnail">

<img src="kittens.jpg" alt="占位符缩略图">

</a>

</div>

<div class="col-sm-6 col-md-3">

<a href="#" class="thumbnail">

<img src="kittens.jpg" alt="占位符缩略图">

</a>

</div>

<div class="col-sm-6 col-md-3">

<a href="#" class="thumbnail">

<img src="kittens.jpg" alt="占位符缩略图">

</a>

</div>

<div class="col-sm-6 col-md-3">

<a href="#" class="thumbnail">

<img src="kittens.jpg" alt="占位符缩略图">

</a>

</div>

</div>

## 添加自定义内容

可以向缩略图中添加各种HTML内容，比如标题、段落和按钮。具体步骤如下:

1.添加class .thumbnail的<div>

2.在<div>内，添加任何想要添加的东西

代码:

<div class="row">

<div class="col-sm-6 col-md-3">

<div class="thumbnail">

<img src="kittens.jpg">

<div class="caption">

<h3>缩略图标签</h3>

<p>一些文本</p>

<p>

<a href="#" class="btn btn-primary" role="button">按钮</a>

<a href="#" class="btn btn-default" role="button">按钮</a>

</p>

</div>

</div>

</div>

</div>

# Bootstrap 警告(alerts)

警告向用户提供了一种定义消息样式的方式。

1.alert alert-success - 告警成功提示

2.alert alert-info - 告警信息提示

3.alert alert-warning - 警告提示

4.alert alert-danger - 危险提示

代码:

<div class="alert alert-success">成功！很好地完成了提交</div>

<div class="alert alert-info">信息！请注意这个信息</div>

<div class="alert alert-warning">警告！请不要提交</div>

<div class="alert alert-danger">错误！请进行一些更改</div>

## 可取消的警告(alert-dismissable)

创建一个可取消的警告步骤:

1.创建一个<div>，并向其添加一个class .alert和class .alter-success

2.向上面的<div>添加可选.alert-dismissable

3.添加一个关闭按钮

代码:

<div class="alert alert-success alert-dismissable">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">

&times;

</button>

成功！很好地完成了提交

</div>

## 警告(alert)中的链接

在警告中创建链接的步骤：

* 1.创建一个<div>，并向其添加class .alert和class .alert-success
* 2.使用class .alert-link来提供带有匹配颜色的链接

代码:

<div class="alert alert-info">

<a href="#" class="alert-link">

信息！请注意这个信息

</a>

</div>

# Bootstrap 进度条

使用Bootstrap创建加载、重定向或动作状态的进步条

## 默认的进度条

创建一个基本的进度条的步骤:

1.添加一个带有class .progress的<div>

2.添加一个带有class .progress-bar的空的<div>

3.添加带有百分比的宽度的style属性，如style="width:60%"

代码:

<div class="progress">

<div class="progress-bar" role="progressbar" aria-valuenow="40"

aria-valuemin="0" aria-valuemax="100" style="width:40%">

<span class="sr-only">40%完成</span>

</div>

</div>

## 进度条颜色

使用class .progress-bar-\*属性来设置进度条的颜色

1.progress-bar-success - 成功颜色

2.progress-bar-info - 信息颜色

3.progress-bar-warning - 警告颜色

4.progress-bar-danger - 危险颜色

代码:

<div class="progress">

<div class="progress-bar progress-bar-danger" role="progressbar" aria-valuenow="40"

aria-valuemin="0" aria-valuemax="100" style="width:40%">

<span class="sr-only">40%完成</span>

</div>

</div>

## 条纹的进度条(progress-striped)

使用属性class .progress-striped来创建进度条

代码:

<div class="progress progress-striped">

<div class="progress-bar progress-bar-danger" role="progressbar" aria-valuenow="40"

aria-valuemin="0" aria-valuemax="100" style="width:40%">

<span class="sr-only">40%完成</span>

</div>

</div>

## 动画的进度条(active)

在一个带有class .progress和class .progress-striped的<div>中。同时添加class .active属性，如:<div class="progress progress-striped active">

## 堆叠进度条

将多个进度条放在相同的.progress中即可实现堆叠

代码：

<div class="progress progress-striped active">

<div class="progress-bar progress-bar-info" role="progressbar"

aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"

style="width:40%">

<span class="sr-only">40% 完成</span>

</div>

<div class="progress-bar progress-bar-success" role="progressbar"

aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"

style="width:30%">

<span class="sr-only">40% 完成</span>

</div>

<div class="progress-bar progress-bar-danger" role="progressbar"

aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"

style="width:20%">

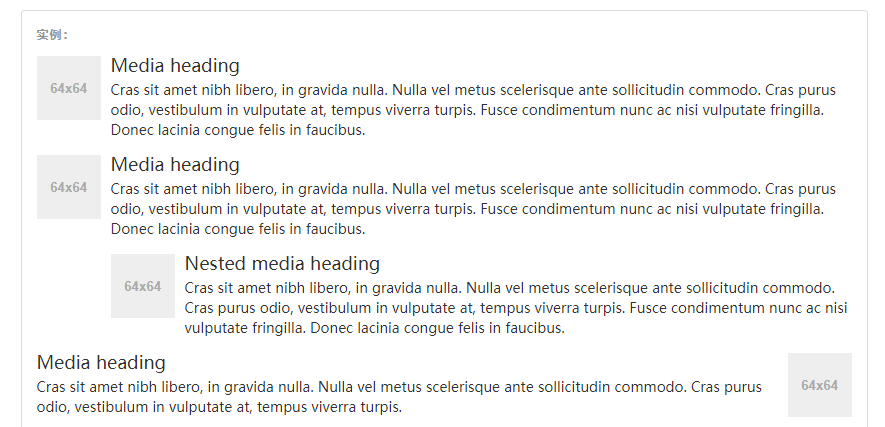
<span class="sr-only">40% 完成</span>

</div>

</div>

# Bootstrap 多媒体对象(media)

效果如下：



使用class .media来设计轻量级的媒体，有下列两个标签

1.media：该class允许多媒体(图像、视频、音频)浮动到内容区块的左边(meida-left)或右边(media-right)

代码:

<div class="media">

<a class="media-left" href="#">

<img class="media-object" src="1.jpg"

alt="媒体对象">

</a>

<div class="media-body">

<h4 class="media-heading">媒体标题</h4>

<span>

示例文本

</span>

</div>

</div>

# Bootstrap 列表组(list-group)

列表组是灵活又强大的组件，不仅能用于显示一组简单的元素，还能用于复杂的定制的内容。

## 基本列表组件实例

1.向元素<ul>添加class .list-group

2.向<li>添加class .list-group-item

代码:

<ul class="list-group col-md-3">

<li class="list-group-item">thinking\_fioa</li>

<li class="list-group-item">ppp</li>

<li class="list-group-item">luweilin</li>

</ul>

## 向列表组添加徽章(badge)

向<li>元素中添加<span class="badge">即可添加徽章。会自动定位到右边

代码:

<ul class="list-group col-md-3">

<li class="list-group-item">

<span class="badge">新</span>

thinking\_fioa</li>

<li class="list-group-item">ppp</li>

<li class="list-group-item">luweilin</li>

</ul>

## 向列表组添加链接/按钮/禁用/情景类

使用<a>来添加链接

代码:

<div class="col-sm-3">

<a href="#" class="list-group-item active">

thinking\_fioa

<span class="badge">帅哥</span>

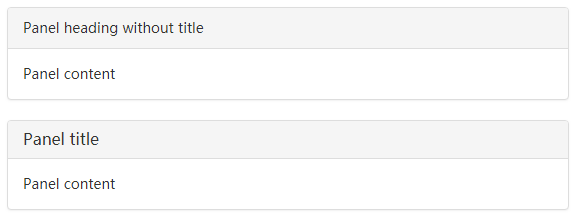
</a>

<a href="#" class="list-group-item">ppp</a>

<a href="#" class="list-group-item">luweilin</a>

</div>

# Bootstrap 面板(panel)



面板控件用于把DOM组件插入到一个盒子中。需要使用class .panel和class .panel-default

代码:

<div class="panel panel-default">

<div class="panel-body">

这是一个基本的面板

</div>

</div>

## 面板标题(panel-title)

可以使用两种方式来添加面板标题:

1.panel-heading - 向面板添加标题容器

2.使用.panel-title的<h1>和<h6>添加预定义样式的标题

代码:

<div class="panel panel-default">

<div class="panel-heading">

不带title的面板标题

</div>

<div class="panel-body">

面板内容

</div>

</div>

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title">

带有title的面板标题

</h3>

</div>

<div class="panel-body">

面板内容

</div>

</div>

## 面板脚注(panel-footer)

使用属性class .panel-footer来添加面板脚注

代码:

<div class="panel-footer">面板脚注</div>

## 带语境色彩的面板

设置语境色彩的面板有: panel-primary、panel-success、panel-info、panel-warning和panel-danger

## 带表格的面板

在面板中使用class .table来创建一个无边框的表格

代码:

<div class="panel panel-default col-md-6">

<div class="panel-heading">

<h3 class="panel-title">面板标题</h3>

</div>

<div class="panel-body">

这是一个基本面板

</div>

<table class="table">

<tr>

<th>姓名</th><th>年龄</th>

</tr>

<tr>

<td>luweilin</td><td>24</td>

</tr>

<tr>

<td>ppp</td><td>25</td>

</tr>

</table>

</div>

## 带列表组的面板

代码:

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title">面板标题</h3>

</div>

<div class="panel-body">

一个基本面板内容

</div>

<ul class="list-group">

<li class="list-group-item">thinking\_fioa</li>

<li class="list-group-item">luweilin</li>

<li class="list-group-item">ppp</li>

</ul>

</div>

# Bootstrap Well

把 Well 用在元素上，就能有嵌入（inset）的简单效果。

使用class .well属性即可

尺寸大小(well-lg、well-sm)

1.well-lg:尺寸大

2.well-sm:尺寸小